natural

Single Point of Development		Introduction

software AG

This document applies to Natural Version 4.1.2 for Mainframes, Version 6.1.1 for UNIX, Version 6.1.1 for Windows, and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Table of Contents

Introducing Natural Single Point of Development - Overview

Introducing Natural Single Point of Development - Overview

This document covers the following topics:

Introducing Natural Single Point of Development

- What is Natural SPoD?
- Remote Development Interface
- Application Concept
- System Architecture
- Natural-SPoD-Related Frequently Asked Questions

What is Natural Single Point of Development?

The following topics are covered:

- Features and Benefits
- One Natural for All

If you have been developing applications using Natural in a non-graphical environment, you have undoubtedly become familiar with Natural's wide range of capabilities and most likely also with some of its inconveniences and shortcomings. You have probably quite often wished for a more advanced user interface for "your Natural". With Natural Single Point of Development, your wish has indeed become reality.

Simply speaking, one might say that Natural Single Point of Development is a Windows-based Natural workstation for remote application development on different platforms.

Actually, Natural Single Point of Development concept covers more than a simple workstation in that it combines the advantages of two disparate worlds:

- the strengths of the character-based Natural products and
- the ease of use provided with Natural for Windows.

In addition, it provides a series of novel features which are explained below.

Features and Benefits

Natural Single Point of Development approach is Software AG's response to the increasing demand for a state-of-the-art development workstation for building, testing and maintaining all Natural applications throughout their life cycle and across all supported platforms.

It meets the following requirements and offers the following advantages:

- **Client/server architecture enabling one single remote development environment for all platforms**

  Natural Single Point of Development is based on a client/server concept. On the client side, Windows-based Natural Studio with its modern look and feel, its powerful drag-and-drop copy and paste functionality and its browser-style workspaces offers one single, uniform view for all Natural users. Its graphical user interface is the basis for a single, uniform working environment for all platforms (mainframe, mid-range and PC computer systems) supported by Natural.

  For more details, refer to System Architecture.

- **Full remote development access after mapping to the target environment**

  For remote development, the Windows-based Natural client (which will remain useable for Windows-only application development) can be enabled to interact with a development server that can be located on a single or multiple, homogeneous or heterogeneous mainframe or mid-range computer systems. Once you have mapped your target environment in the client's browser-style application workspace, you have a set of convenient and effective tools at hand that enable you to perform all development tasks directly in the target development environment. You can use the functionality to the extent it is available there. And to prevent concurrent updates, the object under work on the target platform is locked.

For more details, refer to Remote Development.

- **Single, familiar system image of all objects and ressources involved in application development**

  Natural and non-Natural objects can be accessed and processed with a single user interface.You can generate DCOM components, use RPC and EntireX communications and/or put your applications to the Internet. You can submit and monitor jobs, control job listings and perform dataset maintenance tasks. Operations on all of these different object types, whether they reside on OS/390, VSE/ESA, BS/2000, UNIX (development server already implemented), Windows platforms supported by Natural (development servers are under development or planned), is afforded using a well-known graphical user interface.
  Under consideration: You can display and edit text files, JCL, and code held as Natural objects or 3GL programs.

  For more details, refer to First Steps with Natural Single Point of Development.

- **New application concept giving a logical view to distributed applications**

  To meet the requirements of a new, distributed application development scenario, a new-defined application concept has been introduced which provides a logical view of Natural and future objects on the various sites where they are developed and used. On the workstation screen, this is complemented by an application workspace which shows all distributed objects of an application in a tree structure.

  For more details, refer to Application Concept.

- **Remote development server file**

  A new central data dictionary file has been introduced to cope with the requirements of the new, distributed application concept. Having the same structure as the well-known Natural system file FDIC, the new system file serves to store the information where the objects linked to an application are stored and which objects are locked.

- **Reduced training costs plus increased productivity**

  The use of only one easy-to-use working environment and the independence from operating system or TP monitor will reduce the investment in training otherwise required for the different environments and will shorten the turnaround time for application development.

- **Future-proof by pluggable extras**

  As a first example, XRef Evaluation is available as an optional plug-in unit for the workstation.This facility enables you to retrieve and display conveniently the cross-reference information which is essential for developing applications in Natural. It gives you a comfortable way of listing and navigating through hierarchies of referenced and referencing objects, showing relationships between and within program objects.

  For more details, refer to XRef Evaluation.

  Additionally, plug-ins are available integrating the functionality of the well known products Predict, Natural Construct and Natural Engineer.

  The Object Description plug-in gives access to all descriptive data stored in Predict. A full set of maintenance and retrieval functions is included in the same graphical way. Use of retrieval models and an extended find dialog are included as well.

  The Schema Generation plug-in enables the user to generate DDMs, Adabas files and DB2 tables, views and more.

  Generation of Natural programs becomes as easy as possible, using the Program Generation plug-in. Using a wizard based graphical user interface, the server programs are generated by using Natural Construct running on the server platform.

Other plug-ins are under development. Even third-party plug-ins may be easily integrated in the future. Which plug-ins are actually visible and active can be configured on a per user basis using the Plug-in Manager.

- **Enhanced online help/documentation for remote environments**

  For remote environments with character user interface, Natural Single Point of Development enables application developers and administrators to display context-sensitive help and additional documentation in electronic form using browser style windows within the graphical user interface. This means fast and efficient access to vast amounts of information which formerly required many volumes of printed manuals. Should you need complex information in printed form, you can print out your personal copy of each document that is available online.

  In addition, the syntax help in the program editor available in Natural Studio offers full-detail context-sensitive help for the following Natural syntax elements:

  - Statements
  - System variables
  - System functions
  - Parameters (for example, the AD parameter)

One Natural for All

Natural Single Point of Development is more than wishful thinking - it is indeed **one Natural for all**.

Remote Development Interface

The primary goal of the Single Point of Development (SPoD) approach is to provide a development interface that enables software engineers to develop Natural applications for any platform using only the Windows-based graphical user environment of Natural Studio.

The essential new features are:

- **Remote File Manipulation**

  In the Natural Studio views, you can manipulate (e.g., move, copy) program objects, regardless of location.

- **Remote Editing**

  You can retrieve Natural source files transparently from the target environment, edit them at your workstation and then save them to the target environment. The GUI supported dialogs of the editors provide significant advantages over the character-based editors.

- **Remote Compiling**

  You initiate the compilation process from your workstation by issuing commands to the target environment.

- **Remote Debugging**

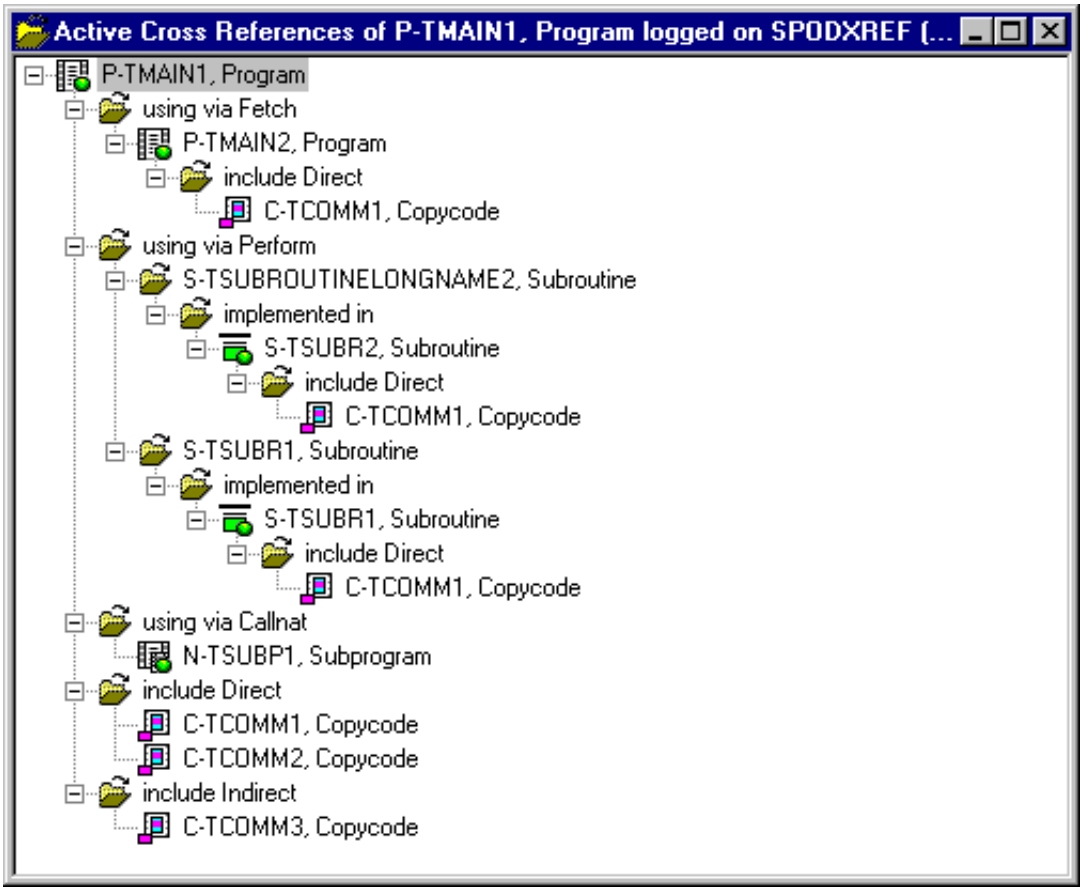
  The debugger available with Natural Studio can be used to debug applications which execute in a target environment that can be located on the same system or in Natural mainframe server environments.

- **XREF Information**

  XRef Evaluation is available as an optional plug-in unit for Natural Studio. It stores cross-reference information created during CAT or STOW commands in a remote FDIC system file. This facility enables you to retrieve and display this essential development and maintenance information conveniently in Natural Studio. This gives you a comfortable way of listing and navigating through hierarchies of referenced and referencing objects, showing relationships (cross-references) between and within program objects. See also XRef Data.

  You need to have Predict installed on the target environment.

  The figure below shows an XRef example screen. For more details and illustrations, refer to First Steps with Natural Single Point of Development.

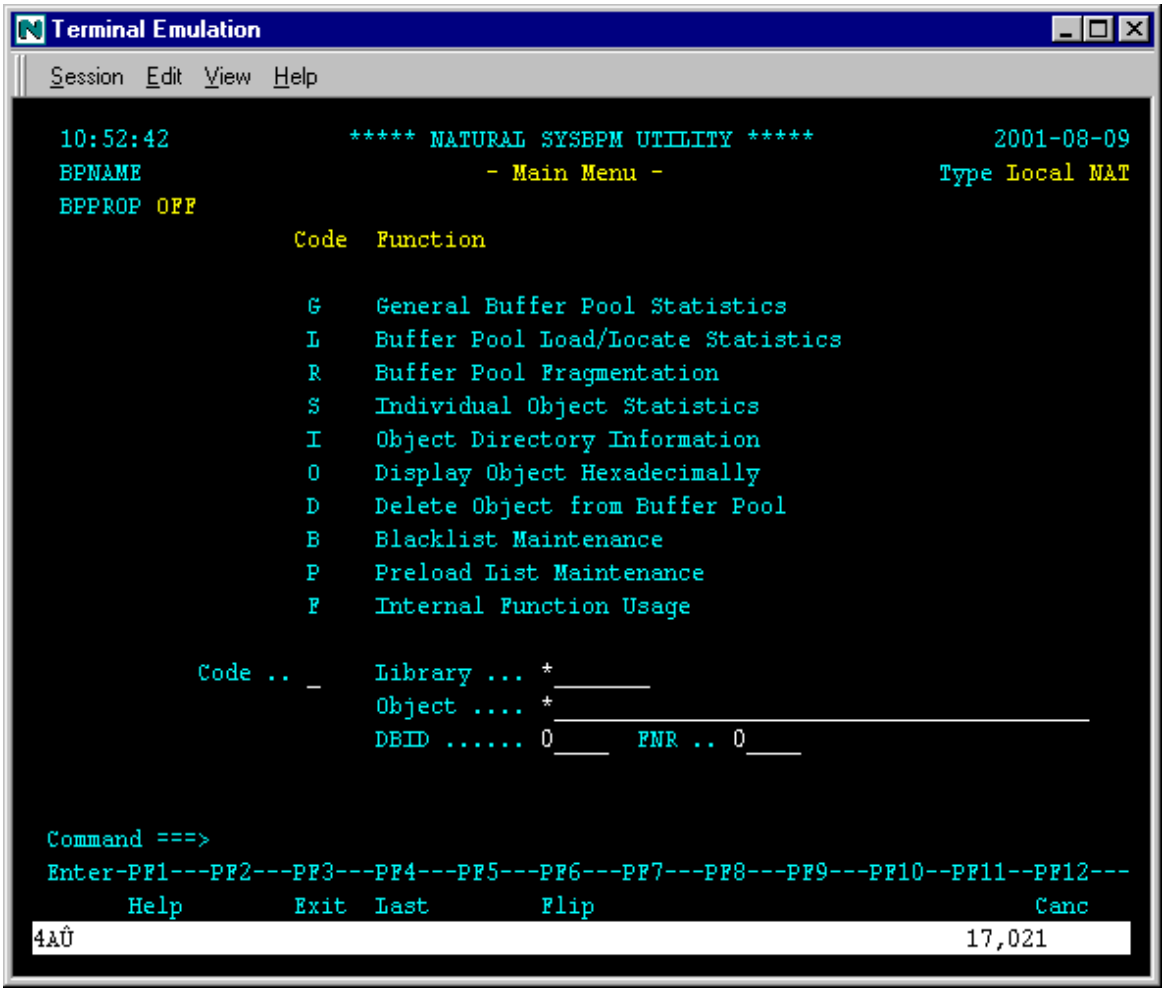


- **Object Locking**

  When you access a remote development server, an object locking mechanism prevents concurrent updates. Locking information is kept in the development server file.

- **Pop-up Terminal Emulation Window**

  Maintenance of mainframe applications often involves testing output to terminals. During remote debugging from Natural Studio, a terminal emulation window pops up automatically for that purpose. This terminal emulation is also available when you are using a utility with a character user interface, for example:



In summary, the benefits of remote development and maintenance are:

- Better control over Natural development tasks
- Increased productivity through a powerful graphical work area
- Lower costs for software development, maintenance and administration
- Greater job satisfaction for software engineers

SPoD Application Concept

The following topics are covered:

- Conventional Approach
- SPoD Approach
- Base Application
- Compound Application
- Application Workspace

Conventional Approach

A conventional Natural application consists of a collection of Natural and Non-Natural objects. Together, they form a functional unit which covers the business logic for a particular business problem. An application consists of a set of libraries and their Natural objects. It is possible that one part of the library belongs to one application while another part belongs to another (different) application.

SPoD Approach

With the SPoD approach, the term "Natural Application" is introduced. It provides a **logical view** of Natural and Non-Natural objects (such as job control files) as well as DDMs. A Natural application does not contain the objects, but it is a **collection of links** to Natural objects or "sub-applications" describing where the objects are stored. Natural objects or "sub-applications" can be linked to several applications.

In a SPoD scenario, the following types of applications exist:

- base application
- compound application

The space where the applications are maintained and displayed at the workstation is called the Application Workspace.

The definitions are stored in the development server file.

Note: Predict Version 4.2 and Version 4.3 support this application concept. Earlier versions of Predict cannot be used.

Some Arguments for Using the Term "Natural Application"

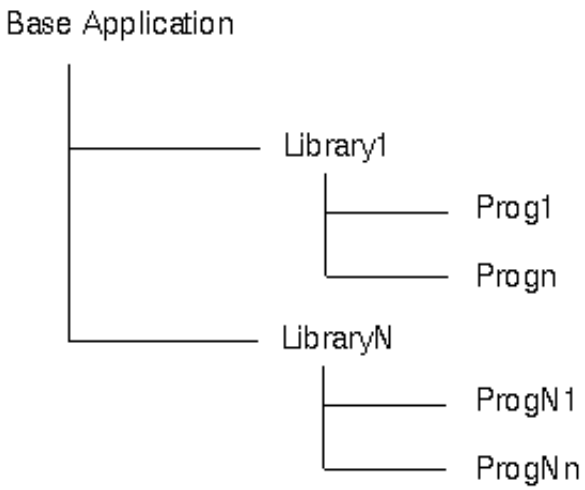
The following topics may give you an idea why you should use the term "Natural Application" and the concept that stands behind it:

- It groups the content of your libraries in a logical order based on the different applications you are maintaining that can be displayed in parallel to the library structure without using an additional tool.
- It focuses your view on the number of objects of a library you intend to work on. You need no longer navigate through all Natural objects located in one library, because you can edit the objects directly from the Application Workspace.
- It reduces the maintenance effort, because you are able to link all objects of different libraries to the application, especially those located in the STEPLIBs.
- It gives you the opportunity to group an application into different "sub-applications" whereas each "sub-application" can be assigned to a member of the development team that is responsible for its development or maintenance.

- Thinking of client/server architecture, you are able to group your application into different "sub-applications", and each "sub-application" can be stored on a different platform at application runtime.

Base Application

A base application is defined as a set of Natural object links. The associated Natural objects pertain to one specific Natural Development Server and are all located on the **same** FUSER system file.



Base Application Structure (Example)

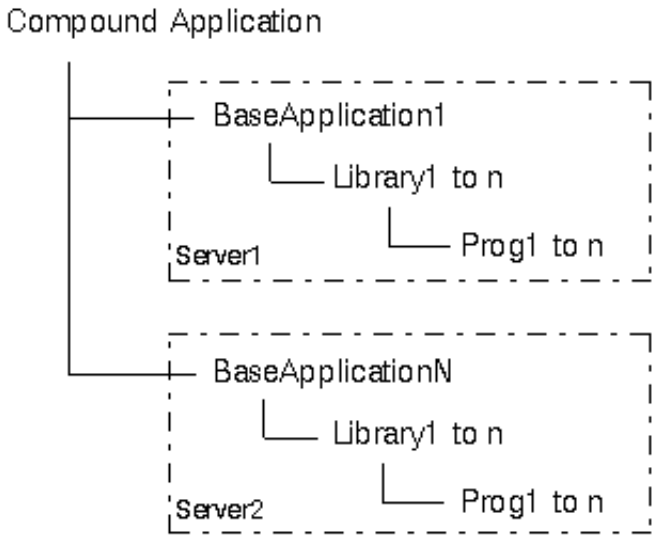
The following information is stored for a base application:

- Name of the application
- Description of the application (textual information only)
- Name and port ID of the Natural development server where the linked objects reside, i.e. the application environment
- Profile for the application environment
- Entry points describing the start objects of your application
- Identification of each object linked

Compound Application

A compound application enables the application developer to combine several base applications. It is defined as a set of links to these base applications.

The base applications involved in a compound application can be spread across **different** FUSER system files or different development servers. Each base application may have a different parameter setting.



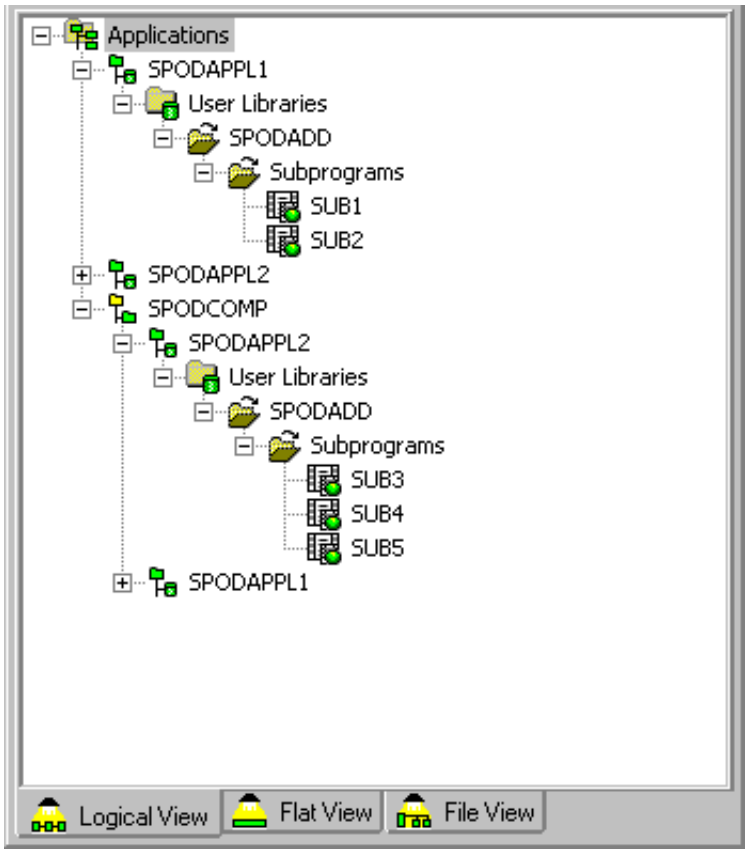
Compound Application Structure (Example)

The following information is stored for a compound application:

- Name of the application
- Description of the application
- Identification of each base application linked

Application Workspace

The Application Workspace is an area on the Natural Studio screen (similar to Natural Studio's Library Workspace) where all known applications and their objects are shown in a view which complements the existing logical, flat and file views and displays a tree structure comprising all program objects belonging to an application, see example below.



SPoD System Architecture

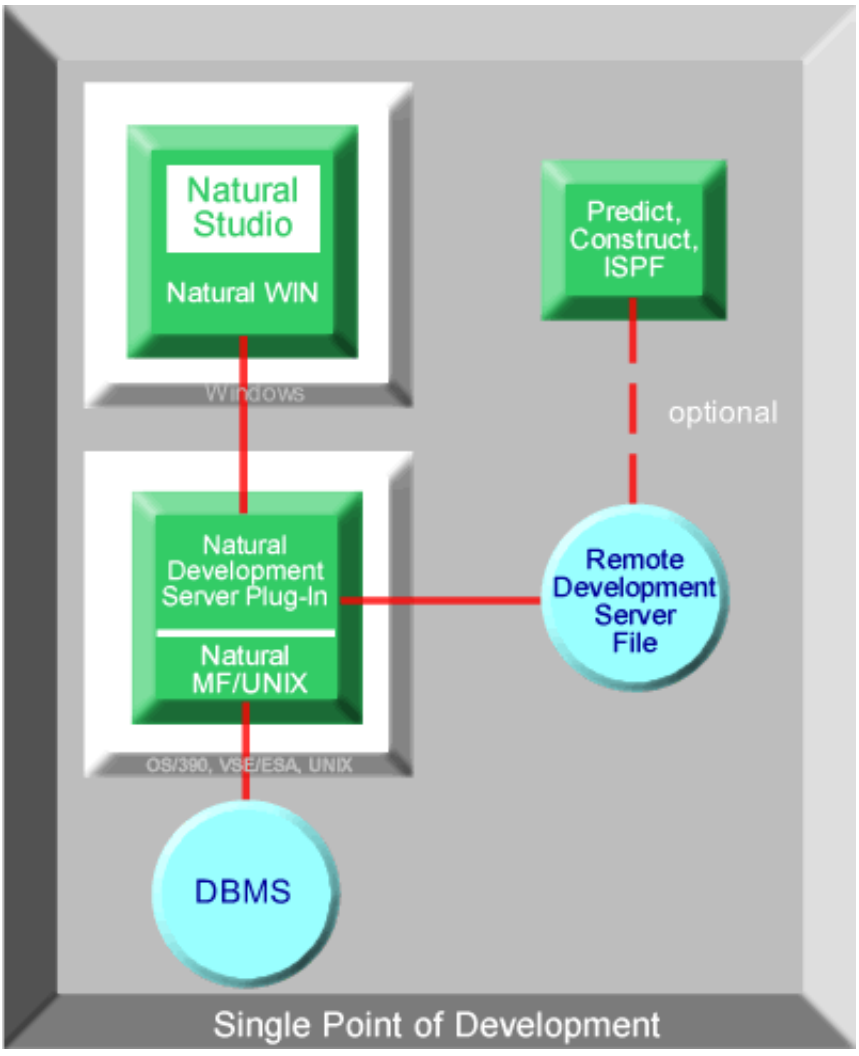
This document describes the Natural Single Point of Development (SPoD) system architecture. The example shown below refers to a remote development server that runs on a mainframe under the operating system OS/390.

The following topics are covered:

- Typical Scenario
- Natural for Windows as Remote Development Client
- Natural Development Server
- Development Server File
- Database Management System
- Predict and Natural Construct

Typical Scenario

A typical Natural SPoD scenario on a mainframe system running under OS/390 is made up of the components shown in the following block diagram:



The details of the block diagram are described below.

Natural for Windows as Remote Development Client

As of Version 5.1, Natural for Windows with its graphical user interface can be used as a remote development client. In this case, Natural Studio is the central development workstation for all platforms.

All the Natural-related steps in the application development cycle including the configuration steps can be done from within Natural Studio, independent of the platform where the specific development environment resides. Remote communication with the platform-specific Natural environment takes place via a Development Server located on the different platforms, involving the use of a protocol and using TCP/IP as physical transport layer.

Most of the functionality needed for remote development (administration, application workspace, remote editing, compiling, debugging, etc.) has been integrated in Natural Studio. For storing cross-reference information, you can use optional plug-in units (not shown in the above figure) providing GUI front ends for XRef evaluation, object description, schema generation and program generation for Natural Studio. These plug-ins require the related products (Predict and Natural Construct) on the server side. Additional plug-in units for other purposes are planned.

Natural Development Server

To enable remote development on a specific platform, e.g. on a mainframe system, a Natural Development Server has to be installed. The development server is basically a Natural running in the Natural server environment in conjunction with the add-on product Natural Development Server (NDV) installed.

Natural Development Servers for OS/390, VSE/ESA and UNIX are currently provided. Base requirement is Natural Version 3.1.6 for Mainframes or above or Natural Version 6.1.1 for UNIX or above installed. Servers for other platforms are planned. For details, refer to the Natural Development Server documentation.

The remote development server covers the following functions:

- provide access to the system files,
- access the application data,
- use the remote development server file to guarantee the consistency of application modifications by object locking, independent of the workspaces used,
- execute the remote commands issued from the development client (Natural Studio).

Development Server File

A Development Server File stores application data and holds object locking information. This file is structured like the Natural FDIC system file and may coexist with an already existing FDIC file. It cannot be modified dynamically or via user exits. It is addressed using the Natural parameter FDIC.

Database Management System

The Natural remote development environment can be used with any Natural-supported database management system that is available for the specific platform and that runs under the existing operating system.

Predict and Natural Construct

The development server file and the FDIC file used by Predict and Natural Construct can be identical, but it is not required. However, if you intend to use your existing Predict or Natural Construct installation, it will be necessary that you migrate to a version as specified under Prerequisites for Natural Single Point of Development.

Natural SPoD Frequently Asked Questions

This document contains frequently asked questions regarding Natural Single Point of Development. Where applicable, reference will be made to relevant problem descriptions/solutions that are maintained in Software AG's Support Information System SAGSIS.

Which versions of Natural are useable in a SPoD environment?

As a prerequisite for **creating and operating** a Natural remote development environment, the following versions of the following products are currently required:

- Natural for Windows Version 5.1.1 or higher
- Natural Remote Development Server Version 1.1.1 or higher
- Natural for Mainframes Version 3.1.5 or higher

Any earlier version of Natural for Mainframes or for Windows (and, in the future, Unix or OpenVMS) can be administered in a SPoD environment. The same applies to the applications and objects created with earlier Natural versions. For remote development, however, only the aforementioned Natural versions can be used.

Which versions of Natural and the related products are required to use the full set of functionality?

As a prerequisite for creating and operating a Natural remote development environment, or for using additional SpoD functionality, the following versions of the following products are required:

Functionality	Prerequisite
Remote development on OS/390	• Natural Version 5.1.1 for Windows or above • Natural Remote Development Server Version 1.1.1 or above • Natural for Mainframes Version 3.1.5 or above
Remote development on VSE/ESA	• Natural Version 5.1.1 for Windows or above • Natural Remote Development Server Version 1.1.4 or above • Natural for Mainframes Version 3.1.5 or above
Remote development on UNIX	• Natural Version 6.1.1 for Windows or above • Natural Remote Development Server Version 2.1.1 or above • Natural Version 6.1.1 for UNIX or above
Using Xref Evaluation plug-in	• Natural Version 6.1.1 for Windows or above • Natural Remote Development Server Version 2.1.1 or above • Natural Version 3.1.6 for Mainframes or above or Natural Version 6.1.1 for UNIX or above • Predict Version 4.3.1 or above
Using Object Description plug-in	
Using Schema Generation plug-in	
Using Program Generation Plug-in	• Natural Version 6.1.1 for Windows or above • Natural Remote Development Server Version 2.1.1 or above • Natural Version 3.1.6 for Mainframes or above or Natural Version 6.1.1 for UNIX or above • Predict Version 4.3.1 or above • Natural Construct Version 4.5.1 or above

Any earlier version of Natural for Mainframes or for Windows (and, in the future, Unix) can be administered in a SPoD environment. The same applies to the applications and objects created with earlier Natural versions. For remote development, however, only the aforementioned Natural versions can be used.

Can Natural objects of previous versions be used in a SPoD environment?

Yes, Natural objects of previous versions can be used in a SPoD environment without migration.

Do I need Predict?

No, although the SPoD approach requires a Development Server File which is a central Natural system file structured like the FDIC file. This does not mean that Predict is a prerequisite for remote development. Nevertheless, Predict Version 4.2 and subsequent versions will support the SPoD application concept. Predict is needed to use the XRef functionality, the object description and/or the schema generation functionality.

Can I use earlier versions of Predict in conjunction with SPoD?

Definitely not! Using a Predict version lower than 4.2 in conjunction with the remote development environment would destroy the information that is held in the Development Server File. Predict Version 4.2 offers a migration path from the Predict 4.1 FDIC to the Development Server File Version 1.1 so that Predict data can be used in Natural SPoD's Application Manager Version 1.1. Predict Version 4.3 offers a migration path from Predict 4.2 to the Development Server File Version 2.1 so that Predict data can be used in Natural SPoDŸs Application Manager Version 2.1.

Can I use Predict Case in a SPoD environment?

Predict Case (PCA) will not be integrated into the SPoD concept. It can be invoked like any other mainframe Natural application via the terminal emulation feature of the remote development environment. However, Predict Case objects will not appear in the remote development tree view.

Can I use Natural Construct in a SPoD environment?

Yes. Natural Construct (CST) for Mainframes and for UNIX are integrated into the SPoD concept. It can be invoked using the Program generation plug-in. Up to now, the models most frequently used have been included. In future versions of SPoD additional models will be available.

Can I use Predict Application Control in a SPoD environment?

Predict Application Control (PAC) is not yet integrated into the SPoD concept. Currently, it can be invoked like any other mainframe Natural application via the terminal emulation feature of the remote development environment. However, PAC objects will not appear in the remote development tree view. A partial integration of PAC functionality is currently under consideration.

Can I proceed without using a SPoD environment?

Yes, you can use the new SPoD-enabled versions of Natural products (Natural 5.1.1 for Windows, Natural 3.1.5 for Mainframes, Predict for Mainframes 4.2.1, etc.) in the conventional, platform-specific way. Thus you can profit from changes/enhancements made to these versions.

You can decide to change over to the Natural SPoD scenario later. All you need to do is to install the remote development server plug-in (NDV) for the target platform and the desired plug-in(s) for Natural Studio, for example XRef Evaluation.

There will be certain features in Natural Studio (for example, the application view) which you do not need in a Natural development environment for a single platform. In case of doubt, read the help texts. SPoD-specific functionality will be marked accordingly.

What is the difference between the SPoD application concept and the existing application shell concept in Natural for Windows?

The application shell is used with the frame gallery for developing, administering, and monitoring application solutions. Together, they provide an infrastructure in which numerous standards are defined and are implemented in the form of reuseable components, such as dialogs and procedures.

The application concept introduced with Natural Single Point of Development is intended to address the structure of customer solutions so as to get a new **logical** view on the solution which is not limited by the boundaries of the Natural libraries.

